UNIVERSITY OF
ILLINOIS LIBRARY
AT URBANA-CHAMPAIGN
STACKS

CENTRAL CIRCULATION BOOKSTACKS

The person charging this material is responsible for its renewal or its return to the library from which it was borrowed on or before the **Latest Date** stamped below. **You may be charged a minimum fee of $75.00 for each lost book.**

Theft, mutilation, and underlining of books are reasons for disciplinary action and may result in dismissal from the University.

TO RENEW CALL TELEPHONE CENTER, 333-8400

UNIVERSITY OF ILLINOIS LIBRARY AT URBANA-CHAMPAIGN

JUN 05 1998

When renewing by phone, write new due date below previous due date.

L162

Digitized by the Internet Archive
in 2011 with funding from
University of Illinois Urbana-Champaign

http://www.archive.org/details/methodologyforde593chan

#593 44 pa.
$2.20

Faculty Working Papers

A METHODOLOGY FOR THE DESIGN AND ANALYSIS OF INFORMATION SYSTEMS WITH RESPECT TO MULTIPLE CRITERIA

John S. Chandler, Assistant Professor,
Department of Accountancy

#593

College of Commerce and Business Administration
University of Illinois at Urbana-Champaign

FACULTY WORKING PAPERS

College of Commerce and Business Administration

University of Illinois at Urbana-Champaign

August 10, 1979

A METHODOLOGY FOR THE DESIGN AND ANALYSIS OF INFORMATION SYSTEMS WITH RESPECT TO MULTIPLE CRITERIA

John S. Chandler, Assistant Professor,
Department of Accountancy

#593

Summary:

An information system can be viewed as symbiotic relationship between the users of the system and the system itself. Ideally, an information system should be designed and analyzed with equal consideration given to both user constraints and to system constraints. Current approaches, however, concentrate on either the user or the system sides, but not on both simultaneously. The methodology described in this paper provides the designer/analyst with a framework for gaining insight into information system performance from both user and system viewpoints by establishing a causal relationship between user goal attainment and system activity. The developed methodology produces not only measures of current performance, but also new, predictive measures of future performance.

The methodology is based on a multiple goal programming (MGP) formulation of the information system design evaluation problem. This paper describes the derivation of the formulation and applies the resultant formulation to an example system. The interpretation of the measures produced by this methodology are described and then employed in the analysis of the example system. This paper concludes with a discussion of the flexibility of the methodology to be applied to general systems problems and of the future areas of research.

Table of Contents

I.	Introduction to Problem Area	1
II.	Overview of Methodology	5
III.	Goal Programming	8
IV.	MGP Formulation of Information System Design Problem	11
A.	Basic Performance Equation	11
B.	Derivation of the MGP Formulation	16
V.	Example of Model Formulation	22
VI.	Interpretation and Use of Methodology	26
A.	Current Design Evaluation	26
B.	Design Alternative Identification	27
C.	Design Alternative Evaluation	28
VII.	Analysis of Example System	30
VIII.	Summary and Extensions	37
IX.	References	40

List of Figures

Figure 1.	Overview of Methodology	6
Figure 2.	View of an Information System	12
Figure 3.	Sequence of Services	13
Figure 4.	Sequence of Activities	13
Figure 5.	Relationship Between Goal Programming and Information System Characterization	19
Figure 6.	Model of Example System	23
Figure 7.	Results of Design Decision Sequence	35

List of Tables

Table 1.	Feasibility Constraints	20
Table 2.	Raw and Calculated Data for Example System	32

I. Introduction to Problem Area

A continuing concern of the information system community is that there have been two separate approaches to the evaluation of information systems' performance and design--one, which is focused on the computer system domain and the other, which is directed at the user domain. Each has its own goals and measures: in the computer system domain, resource queueing, utilization and cost statistics are accumulated to measure performance while in the user domain, throughput, reliability and response time measures are employed to evaluate the performance of requested services. A workshop sponsored by ACM and NBS[1] concluded that any performance analyses "should recognize both the costs of a computer installation and the needs of users for service."

The complexity of the design problem for modern computer-based information systems has increased significantly over its predecessors. This has been due to many factors such as:

a. an expanding range of users and applications with a corresponding expanding set of diverse performance goals and resource requirements,

b. an increased dependence of system behavior on subtle design decisions and changes in user load mix, and

c. a growing demand to achieve conflicting performance objectives, e.g. time vs. cost.

---

[1] This Workshop was one of a series sponsored jointly by ACM and NBS to examine the major issues involving computers. Performance evaluation was chosen as the topic of this Workshop because of its significant impact on computer usage. A summary of the conclusions appears in BOE75.

In this operating environment, it is quite possible to design a system such that good performance for one or more users is gained at the expense of others. Furthermore, because system resources may be used by different users, improving the performance characteristics of one or more resources for the benefit of specific users may have a detrimental effect on overall performance. The problem presented to the designer is to configure a system which satisfies the users' effectiveness criteria while simultaneously achieving multiple system resource related performance criteria.

Current evaluative techniques do not focus on the complete system design problem. The ability to simultaneously ascertain the impact of resource performance on user goals or vice versa is not readily available through these methodologies.

User oriented analyses with objective functions based on response time, throughput, and cost have been (and are continuing to be) reported in the literature (ADI72, BAS75, BUZ73, GAV76, KLE76). Most of these approaches use an analytical model. Through these models one is able to gain insight into the general behavior of systems under various conditions at minimum cost. In order to maintain tractable models, however, many simplifications are required for analytical solution. As a result, these approaches, in many cases, are not able to identify the relationship between users and resources.

Alternatively, performance analyses can be made from the system's standpoint, treating the user and his goals in the aggregate. The most common approach is a subsystem study, where a particular part of the information system is isolated, with the subsystem (user)s

represented by a stochastic generator, either analytic (ABA68, GOT73, HEL70) or simulative (BOU72, NAH74). These system-based evaluations provide an excellent framework for gaining insight into the detailed local behavior of information systems. They have limited use, however, in evaluating multiple criterion problems because they ignore the impact of the individual user on the operation of the system, preferring to characterize users in the aggregate.

Examinations of total systems have also been made. Many approaches have been employed: accounting data analysis (COO72, MER74), hardware/software monitoring (BAR71, COL72, LIN76, SVO76) and simulation modeling (NOR71), especially of data base systems (REI76, ROS74, SIL74). Although they can produce accurate measures of user performance goals (i.e. response time) and accurate measures of systems' goals (device queueing and utilization), there is little attempt to relate the two causally. But from practical experience it is evident that there is indeed such a relationship. In fact, fundamental laws of computer performance have been proposed that relate resource activity to user measures (BUZ76).

There have also been efforts to evaluate systems from a global view, combining both user and system criteria. Techniques for such analysis include aggregate measures (KAC74), utility functions (GRO72, HAL74, JOS68) and graphical evaluation via Kiviat graphs (KOL73, MOR74). Probably the most comprehensive and ambitious attempt to date has been the ISDOS project (Information System Design and Optimization System). It was begun in the 1960's (TEI67) and since has been extended in several areas (NUN71, NUN76, SEV72). In order for

these approaches to achieve results, however, they simplify the complexity of the multiple criterion problem to such an extent that only a single criterion remains. Thus, the current evaluative approaches are unable to anlayze the complete scope of the design problem defined in this paper.

II. Overview of Methodology

The purpose of the methodology presented here is to provide a mechanism for the designer/analyst to gain insight into information system performance with respect to both user and system criteria. A causal liaison is established between user goal attainment and system activity. As a result, the developed techniques produce measures of current performance and new, predictive measures of future performance, while also providing a framework for further user/system analysis.

The methodology described in this paper is based on three interconnected stages as shown in Figure 1. System design is viewed as being iterative, with each iteration involving the sequential invocation of these three stages for the purpose of improved system performance at the end of each iteration.

A brief summary of each stage in this methodology is presented below.

Stage 1: System Evaluation

This stage evaluates the behavior of a specific information system with respect to system measures. The outputs of Stage 1 are performance statistics for the resources in the aggregate and for their behavior with respect to identified users.

Stage 2: User Goal Evaluation

This stage has two purposes, the first to ascertain the degree of user goal achievement and the second to determine a set of guidelines for altering the current system configuration in order to more

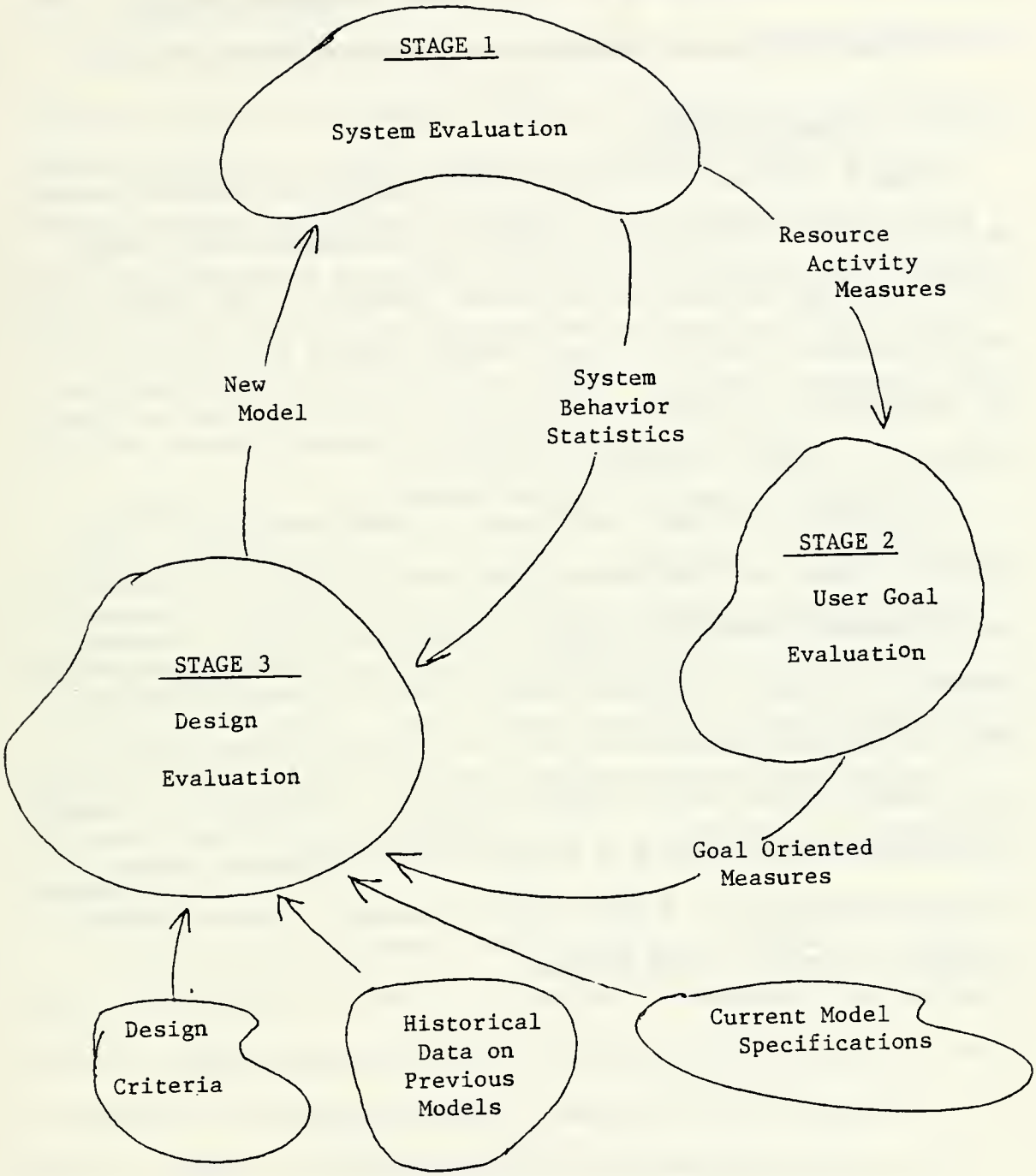

Figure 1 Overview of Methodology

closely attain user goals in the next design iteration. Multiple Goal Programming (MGP) is used as a base for the techniques in this stage.

Stage 3: Design Evaluation

Stage 3 also has two functions. The first is to ascertain whether or not the current design's performance is satisfactory with respect to both the user criteria and the system criteria. If the design is not satisfactory, then this stage's second function is to define a new system based upon the current design, prior alterations, and the results of the Stage 1 and Stage 2 analyses.

The focus of this paper will be on the development of the Goal Evaluation Stage the formulation of MGP based procedures to evaluate the information system with respect to multiple criteria, the establishment of a formal statistical liaison with the System Evaluation Stage, and the interpretation of the Goal Evaluation Stage outputs with respect to their use in Design Evaluation. Although this research is directed towards analysis of computer based information systems, it is also applicable to a broad range of systems, within and without the realm of computer based systems.

III. Goal Programming

In the problem addressed in this paper, the designer must be constantly aware of the impact of design decisions on both user and system criteria and attempt to obtain an equitable balance between them for overall satisficed performance. This design decision falls under the category of a multilple objective decision making situation. There are many approches to this problem, but MacCrimmon (MAC73) has classified them into the four major types: weighting, sequential elimination, spatial proximity and mathematical programming.

Several of these approaches have been applied to information systems: comparative system selection by weighting (JOS68, SCH72), evaluation of time-sharing systems through graphical spatial proximity methods (GRC72), and linear programming models of system performance (SHW65, KAC74). Each of these studies has limited applicability to the multiple criterion design problem at hand because of linear restrictions or dependence on an a priori set of design alternatives.

Multiple goal programming (MGP) is employed for several reasons. First, the objective of MGP is to determine a satisficed, not optimal, solution, which recognizes the set of inherently conflicting goal requirements of informations systems. Second, MGP can characterize not only linear, but also ordinal goal relationships which are also inherent in information system evaluations and which provide for easier model formulation. And third, overall figures of merit as well as individual goal measures can be produced.

Charnes and Cooper (CHA61) were the first to formulate multiple goal programming models in order to solve linear programming problems that, because of conflicting constraints, were infeasible. Ijiri (IJI65), in applying MGP to accounting problems developed a generalized inverse approach. A computer model, based on the Simplex approach was first reported by Lee (LEE72). The applications of MGP continue to increase. Representative efforts concern media planning (CHA68), manpower planning (CHA70, CHA76), LEE72, PRI74), production planning (LEE71, JAA69) and CPA firm management (KIL73). The use of MGP in information system design, however, represents a new application of this technique.

The MGP formulation employed in this paper is based on the approach taken by Charnes and Cooper (CHA61) and Lee (LEE72). The basic formulation is:

.A. Minimize $\lambda = \sum_{i=1}^{N} D(i)$

subject to

$$A \cdot \beta + D = G$$

$$\beta \geq 0$$

Where

A is a (NxM) matrix of technological coefficients
G is a (Nx1) vector of goals
D is a (Nx1) vector of discrepancies
β is a (Mx1) vector of decision variables
M is the number of decision variables
N is the number of goals

The object of an MGP procedure is to minimize the objective function by, essentially, driving the values of the discrepancies as close to zero as possible through manipulation of the values of the decision variables. The result is not an optimal solution like linear

programming, but a satisficed solution, one in which the trade-offs between achievement and non-achievement are minimized. The final values of β represent the levels of the decision variables required to satisfice the stated goals.

To express the general multiple objective situation, one must modify formulation .A. in several ways. First, to allow discrepancies to be unconstrained in sign and to be able to characterize both over and under goal achievement, replace $D(i)$ with $D(i)^- - D(i)^+$ (note: $D(i)^- \cdot D(i)^+ = 0$ and that vector D is replaced by vectors $D^- - D^+$). Second, to accomodate unequal importance of discrepancies, each discrepancy can be weighted individually, i.e. $P(i)^+ \cdot D(i)^+$ and $P(i)^- \cdot D(i)^-$. And third, in order to characterize non-linear, ordinal relationships between goals, MGP provides for preemptive factors to be applied to discrepancies, $F_k^{\pm}(i)$ where i the goal and k is the preemptive priority level. These priority levels form a goal hierarchy which MGP attempts to satisfy, starting at the highest priority level and proceeding down, one level at a time. The final MGP formulation is, thus,

.B. Minimize $\lambda = F^+ \cdot P^+ \cdot D^+ + F^- \cdot P^- \cdot D^-$

subject to

$$A \cdot \beta + D^- - D^+ = G$$

IV. MGP Formulation of Information System Design Problem

IV.A. Basic Performance Equation

The view of an information system assumed in this paper is illustrated in Figure 2. The user makes a request to the system and the system responds by satisfying that request. This response is compared against user-oriented performance goals by the user to evaluate the system. When a user's request enters the information system, it is mapped into a sequence of system services. The specific services in this sequence will perform those information system processes necessary to satisfy the request. An individual service may perform at any level of complexity; application level, data management level or operating system level.

Basic to this discussion is the assumption that the measure of user request performance can be expressed as the linear summation of the performance of a selected set of subordinate system services. The members of this set are called activities and are characterized as aggregations of other system services. Since the concept of a service, as used in this paper, can apply to any level of complexity, an activity is also considered to be a service. The difference lies in the fact that for a given request, service, the sequence of system services initially defined is now partitioned into a non-overlapping linear sequence of activities. As Figures 3 and 4 demonstrate, the initial sequence contianing several services operating in parallel has now been linearized by aggregating several system services into one activity (e.g. S2, S3 and S4 are now Activity 2). Thus, the user service is now characterized by Activities 1 through 4 instead of services 1 through 10.

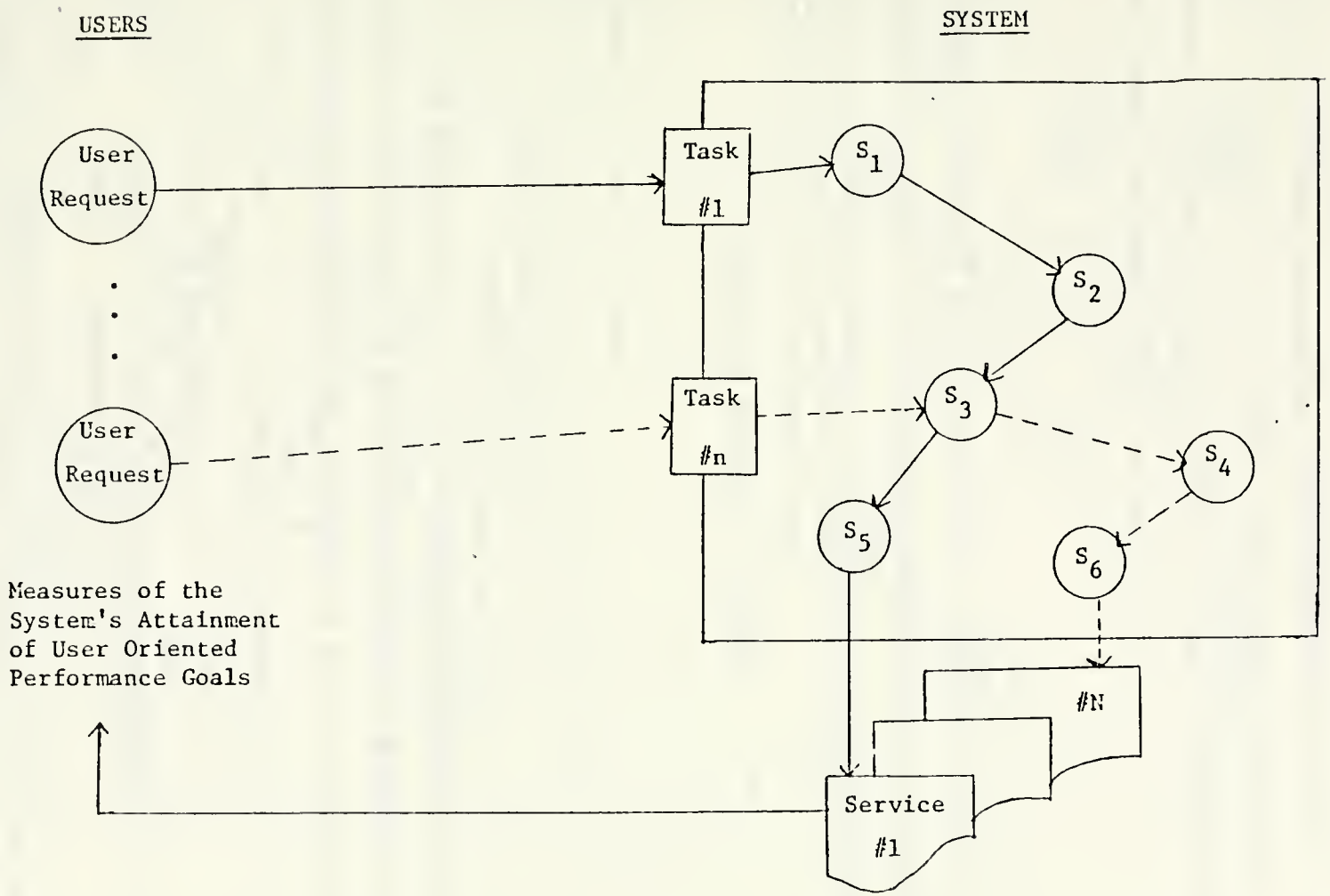

Figure 2: View of an Information System

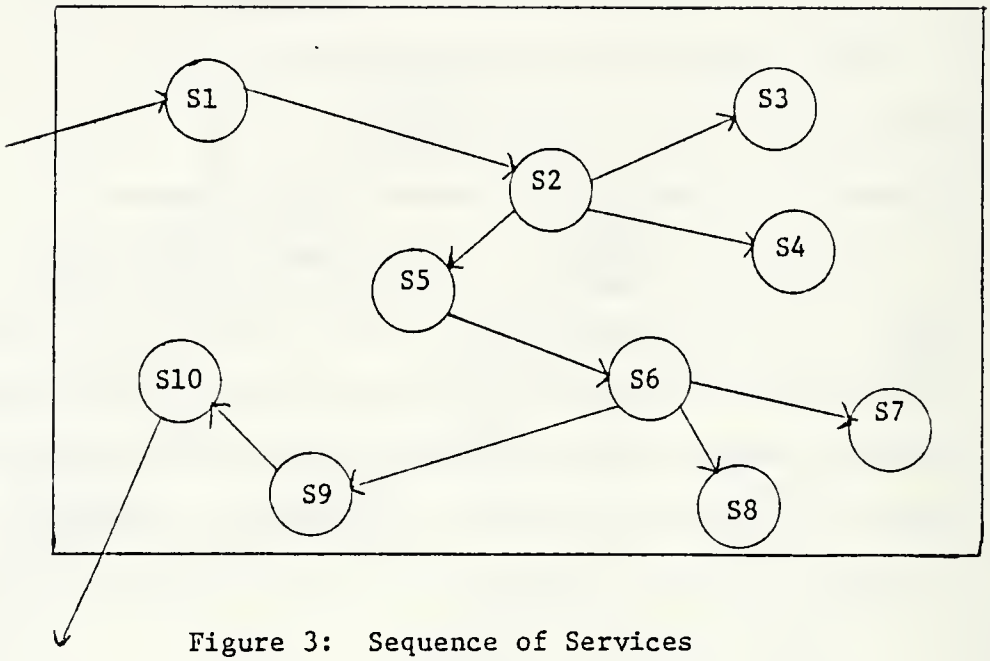

Figure 3: Sequence of Services

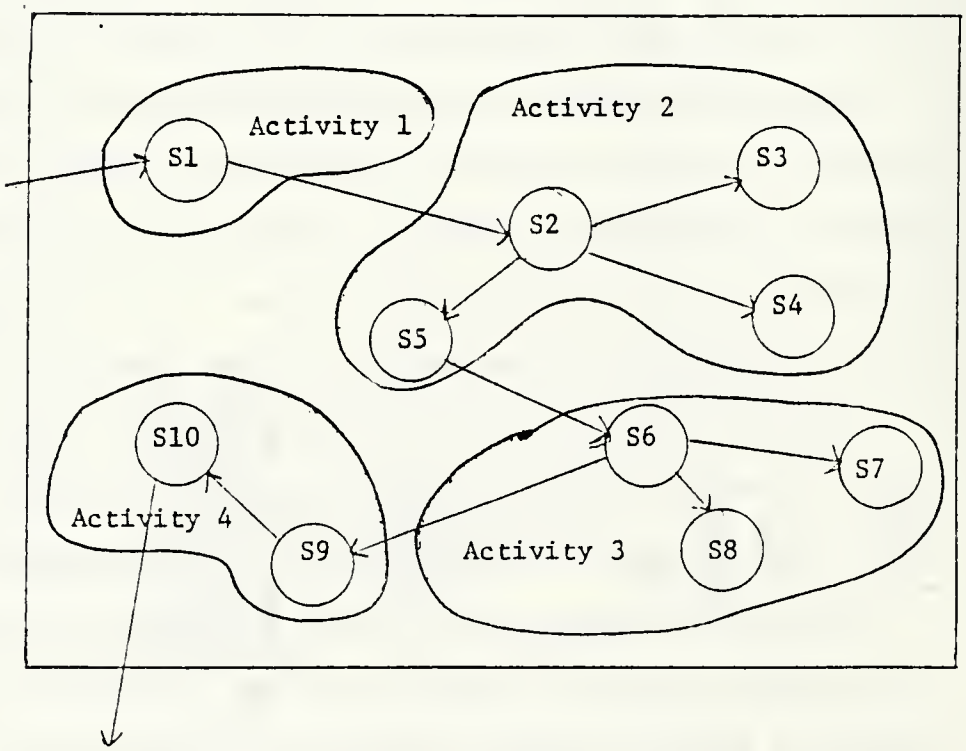

Figure 4: Sequence of Activities

Assume that a given user request i is characterized by a sequence of activities, A_1, A_2, ..., A_n. Let the performance of one of those activities, A_j, with respect to the associated invoking user request, be denoted as $P(A_j(i))$. Thus, the linearity assumption for user request performance can be expressed as:

$$(1) \quad P(i) = \sum_{j=1}^{N} P(A_j(i))$$

For each user goal, a user performance goal, G(i), is assumed to be identified, against which the system's performance is to be compared. The measure of the achievement of this goal is the difference between the goal and the actual performance level,

$$(2) \quad D(i) = G(i) - P(i).$$

Combining (1) and (2) yields

$$(3) \quad D(i) = G(i) - \sum_{i=1}^{N} P(A_j(i)).$$

This value for $P(A_j(i))$ is only for one invocation of the user request service sequence. For this methodology, the expected value over all invocations is employed and is denoted as $R_j(i)$. Let $R_j(i)$ be defined as

$$(4) \quad R_j(i) = \sum_{k=1}^{Nji} P(A_j(i))k/Nji$$

where Nji = the number of invocations of activity j for user service i.

Substituting $R_j(i)$ into (3) for $P(A_j(i))$ yields a general measure of the performance of user service i, over all invocations of that service:

$$(5)\quad D(i) = G(i) - \sum_{j=1}^{N} R_j(i).$$

The significance of this formulation is that it establishes a direct measurable relationship between user-oriented performance, $D(i)$, and system-oriented performance, $R_j(i)$. Since activities are assumed to be aggregates of services, $R_j(i)$ is not only a measure of the activity's performance, but it is also a measure of its subordinate services' performance. Thus, the impact of user demand on the system can be mapped down to the information system services, where the designer is assumed to have design control.

IV.B. Derivation of the MGP Formulation

The MGP formulation is based on the general measure in equation (5). From the discussion in section III, we can replace D(i) with $(D^-(i) - D^+(i))$, and rearranging yields,

$$(6) \quad \sum_{j=1}^{N} R_j(i) + D^-(i) - D^+(i) = G(i).$$

The objective of performance evaluation and redesign would be to minimize these discrepancies, since they represent over/under achievement of a user goal. Since the minimization of such discrepancies is the basis of MGP, the initial objective function is:

$$(7) \quad \text{Minimize } \lambda = \sum_{i=1}^{M} (D^+(i) + D^-(i)).$$

The constraints should be constructed so that the solution of the MGP formulation, i.e. the final values of the decision variables, will yield information for improving performance. Since we have assumed that the designer has control over the design of the activities, the decision variables should relate to them. The current performance levels of activities, the $R_j(i)$'s, in equation (6) provide an evaluation of the current system. With respect to analysis of the current system for potential future performance improvement, those levels may need to change. Thus, the value of a decision varible should reflect the amount of change required.

Let β_j be the decision variable associated with each activity A_j. It is interpretted as the performance change indicated for activity A_j to improve performance of λ (e.g. $\beta_j = .75$ implies a 25% reduction is required). The values of the β_j's are determined by the MGP

procedure as it attempts to minimize $(D(i)^- - D(i)^+)$. Thus, the basic MGP constraint equation for one user request i is:

$$(8) \quad \sum_{j=1}^{N} (R_j(i) * \beta_j) + D(i)^- - D(i)^+ = G(i).$$

Performance evaluation of a total information system is not for one user only, but for all users. Furthermore, a given activity may be in more than one request sequence. Thus, equation (8) must be expanded to cover M user request classes as well as N activities. Adding the general weighting and preemptive factors, the resultant formulation is:

.C. Minimize $\lambda = F^+ \cdot P^+ \cdot D^+ + F^- \cdot P^- \cdot D^-$

subject to

$$R \cdot \beta + D^- - D^+ = G$$

where

- R = (NxM) array of current activity performance levels
- β = (Mx1) array of β_j's
- $D^+, (D^-)$ = (Nx1) array of positive (negative) discrepancies
- G = (Nx1) array of user goals
- M = the number of goals
- N = the number of activities

The satisficed solution to this formulation produces values for the β_j's that are interpretted as indications of what the performance level of the associated activities should have been to minimize the discrepancies, i.e., achieve the goals of the system. If $\beta_j > 1$ then the performance level should have been longer, and likewise, if $\beta_j < 1$ then the performance level should have been shorter. It should be emphasized that these interpretations of individual β_j's should be taken in the context of the entire set of β_j values.

The relationship between β_j, $R_j(i)$ and $G(i)$ are graphically depicted in Figure 5. Assume that user request i ($user_i$) maps into the sequence of activities $A_1, A_2, \ldots, A_N$. Activity A_j serves $user_i$ for an average of $R_j(i)$ amount of time per initiation of $user_i$ (assuming time as the metric). This fits into an overall measure of system service time, $T_{S(i)}$, as one of the N $R_j(i)$'s (see ℓ_1 and ℓ_2). ℓ_3 represents the stated goal for $user_i$ and D(i) is the discrepancy with respect to current performance, $T_S(i)$. The value of β_j (in this case less than 1) would indicate that modifying $R_j(i)$ such that $R_j(i)$ $\beta_j < R_j(i)$ would yield a new $user_i$ performance, $\hat{T}_S(i)$. When this new performance measure is compared to G(i), a smaller discrepancy, is found. $\hat{D}(i)$ thus is interpretted as improved performance. A similar demonstration can be made if $\beta_j > 1$.

In order to complete the formulation of the information system design problem, several additional constraints on the range of the β_j's themselves had to be incorporated into the model. These constraints, called feasibility constraints, are described in Table 1. The discrepancies from these constraints are automatically added to the objective function at the lowest priority level, i.e. one below that of the lowest user specified level. The final formulation of the information system design problem in MGP form is:

.D. Minimize $\lambda = F_D^+ \cdot P_D^+ \cdot D^+ + F_D^- \cdot P_D^- \cdot D^-$
$+ F_L^- \cdot P_L^- \cdot L^- + F_U^+ \cdot P_U^+ \cdot U^+$
$+ F_E^+ \cdot P_E^+ \cdot E^+ + F_E^- \cdot P_E^- \cdot E^-$

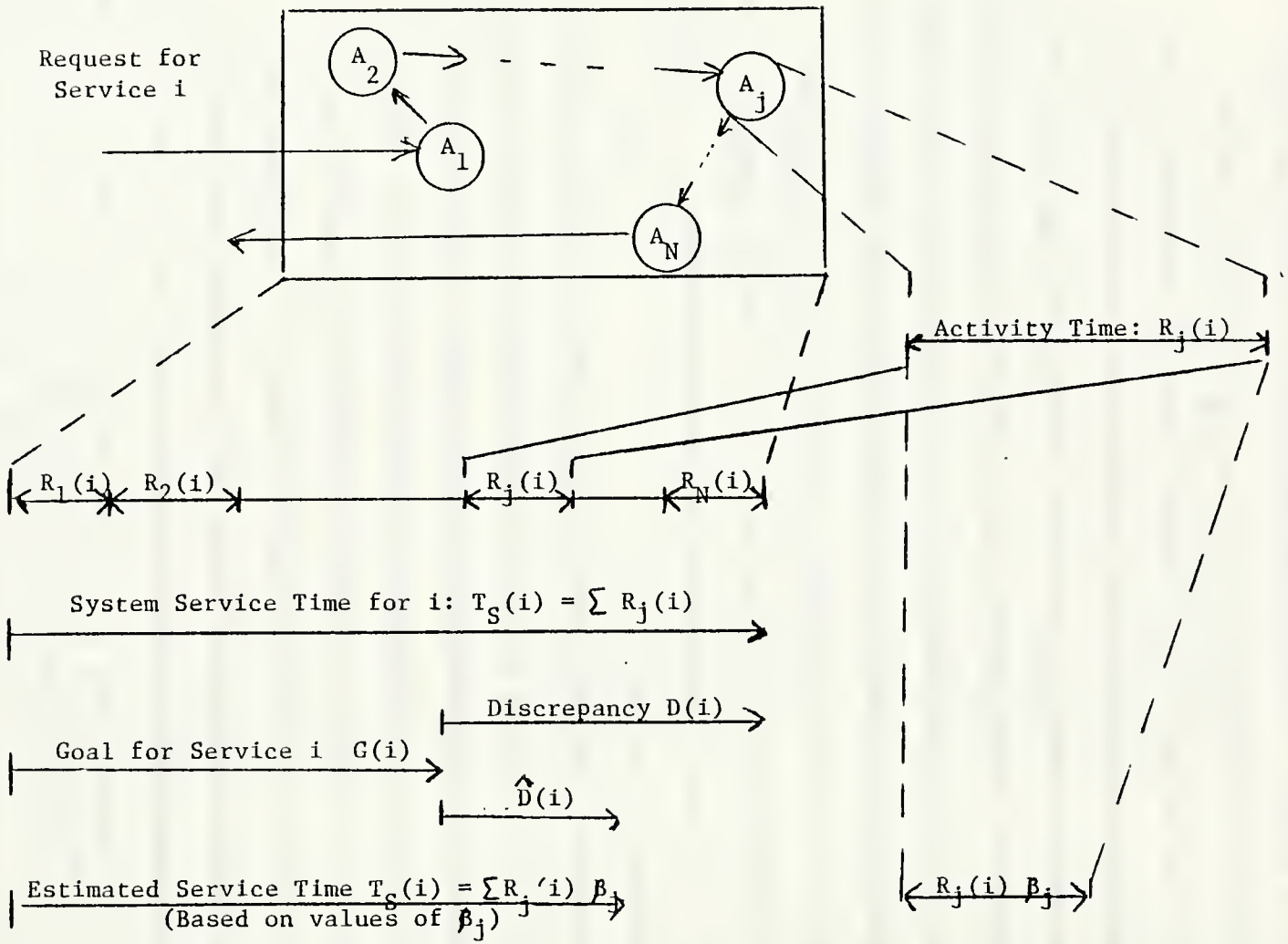

Figure 5 Relationship Between Goal Programming and Information System Characterization

Type of Constraint	Reason for Constraint	Form of Constraint	Vector Definition	Discrepancy included in objective function
Upper Bound	Restrict Solution to technically feasible	$\beta_j + u_j^- - u_j^+ = U_j$ where $1 \geq U_j$	$(u_1^-, u_2^-, \ldots u_n^-) = U^-$ $(u_1^+, u_2^+, \ldots u_n^+) = U^+$ $(U_1, U_2, \ldots U_n) = U$	u_j^+
Lower Bound	Restrict solution to technically feasible: Prevent elimination of an activity	$\beta_j + \ell_j^- - \ell_j^+ = L_j$ where $0 < L_j \leq 1$	$(\ell_1^-, \ell_2^-, \ldots \ell_n^-) = L^-$ $(\ell_1^+, \ell_2^+, \ldots \ell_n^+) = L^+$ $(L_1, L_2, \ldots L_n) = L$	ℓ_j^-
Modification Reduction	Minimize number of β_j's $\neq 1.0$	$\beta_j + e_j^- - e_j^+ = 1$	$(e_1^-, e_2^-, \ldots e_n^-) = E$ $(e_1^+, e_2^+, \ldots e_n^+) = E^+$ $(1, 1, \ldots 1) = \bar{1}$	e_j^-, e_j^+

Table 1: Feasibility Constraints

$$\text{s.t.} \qquad R \cdot \beta + D^- - D^+ = G$$

$$\beta + U^- - U^+ = U$$

$$\beta + L^- - L^+ = L$$

$$\beta + E^- - E^+ = \bar{1}$$

where

R is the matrix of performance levels

β is the array of performance change indicators

G, U, L, and $\bar{1}$ are the arrays of goals for the user criteria and the respective β constraints

$D^{\pm}$, $U^{\pm}$, $L^{\pm}$ and $E^{\pm}$ are the arrays of positive and negative discrepancies from the respective goals

$P_D^{\pm}$, $P_U^{\pm}$, $P_L^{\pm}$ and $P_E^{\pm}$ are the arrays of penalties for the associated discrepancies.

$F_D^{\pm}$, $F_U^{\pm}$, $F_L^{\pm}$ and $F_E^{\pm}$ are the arrays of priority levels for the associated discrepancies

V. Example of Model Formulation

The purpose of this section is to present an example of a model formulation. This example is similar to one of the original models that was used to demonstrate the validity of the methodology (CHN77). The information system to be analyzed is a general, disk-based, online retrieval system, accesing data such as bibliographic abstracts, credit data or inventory levels. The user of this system submits a request to the system, it is analyzed by the system, the appropriate record is retrieved from the data base and then the record is sent back to the user. It is also assumed that this data base system requires an internal control feature of a front-end security screening to prevent unauthorized access to the data base.

The data to be accessed resides in a hierarchic data base (see Figure 6). A credit data base is used in this example. The top level index file (INDEX1) contains an entry for each individual credit item in the data base, referenced by individual name/credit item, e.g. Jones/loan. The second level index (INDEX2) has a corresponding entry for each item in INDEX1 but it references an internal accession number. This number points to the bottom level of the data base which contains the actual credit records. Another file is required for the security function, a list of authorized users (AUTH).

The model of this system follows the structure of these files and indices. Accessing the AUTH file is characterized by a security function (S1), and by a safeguarded, non-shared disk access routine solely for the AUTH file (S5). Accessing the other layers of the hierarchy are characterized by S2, S3 and S4.

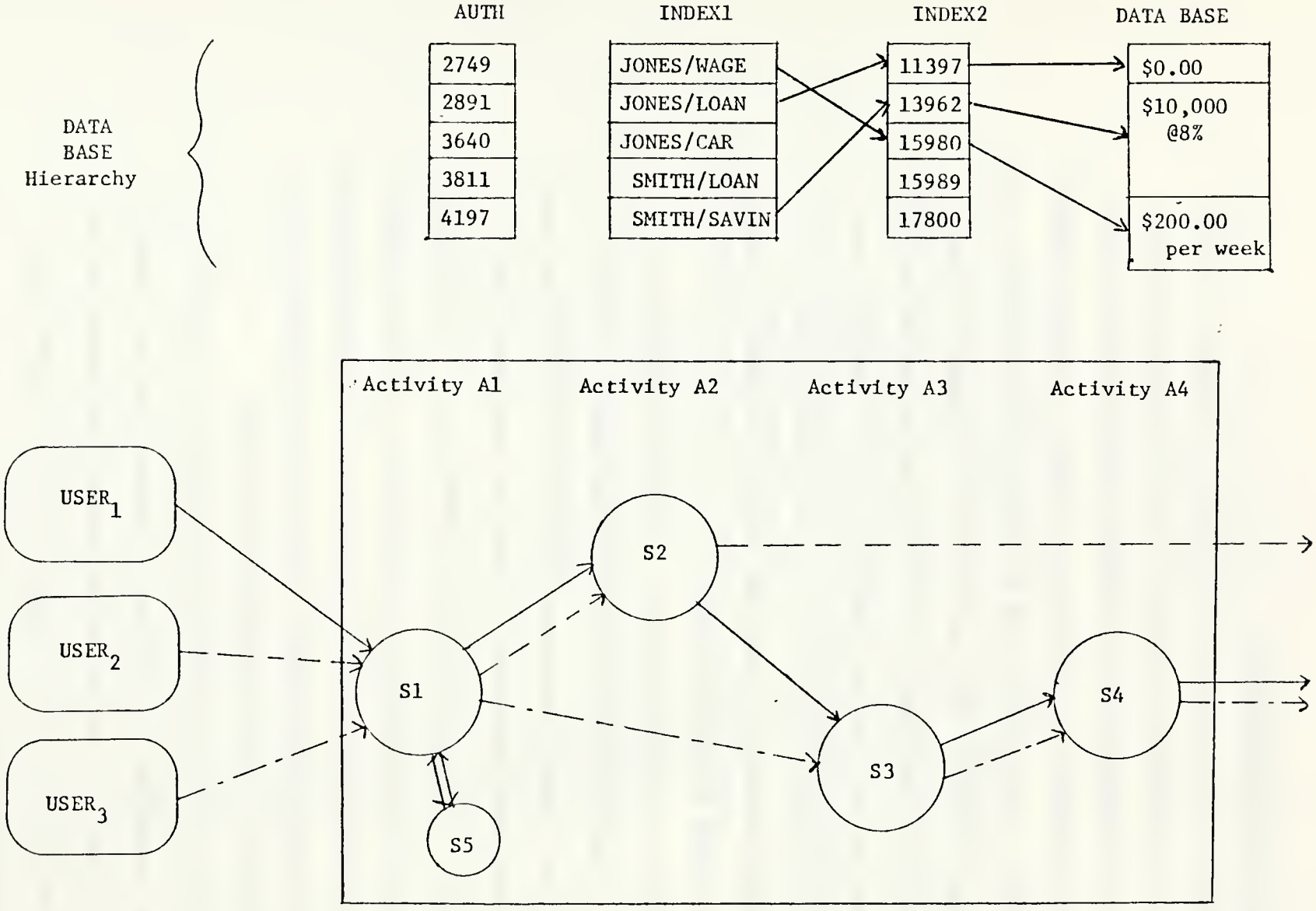

Figure 6: Model of Example System

The processing load against this information system is assumed to come from three types of users. The first user type, USER1, needs to traverse all levels of the data base in search of an individual credit item (S2, S3, S4). As an example, a bank loan officer may need to examine a prospective applicant's bank balance. The second user class, $USER_2$, only has to reference the highest index, INDEX1,to determine the existence of a given credit item in the data base (S2). For example, an information specialist may be examining the growth of the data base and needs to know only the number, not content, of items in the data base. The third user class, $USER_3$, requires the actual credit record, but already has the internal accession number, possibly from a previous access, and can go directly to INDEX2 (S3, S4). All users must go through the security function (S1, S5) and no user can access the actual data without going through an index.

Using the activity creation process described in section IV, four activities can be identified. Since the security disk access function (S5) is performed only at the request of the global security function (S1), one activity, A1, can be defined to be the complete security function. None of the user requests can proceed until this activity is completed. The other invocations of services are also sequential due to the logic and security of the hierarchy. Thus, each service S2, S3 and S4 can be associated with a unique activity A2, A3, and A4. The responses of the system to each type of user request, shown graphically in Figure 6 are:

(9) $USER_1 = f(A1,A2,A3,A4)$

(10) $USER_2 = f(A1,A2)$

(11) $USER_3 = f(A1,A3,A4)$.

In order to transform the information system model above into a form amenable to Stage 2 procedures, the constraints and objective function must be constructed and the goals structure established. It is assumed that each user class, $USER_i$, has a goal, G(i). Then from equations (9) - (11), the user constraints can be formed as follows:

$$(12)\quad R_1 1\ \beta_1 + R_2 1\ \beta_2 + R_3 1\ \beta_3 + R_4 1\ \beta_4 + D(1)^- - D(1)^+ = G(1)$$

$$(13)\quad R_1 2\ \beta_1 + R_2 2\ \beta_2 + D(2)^- - D(2)^+ = G(2)$$

$$(14)\quad R_1 3\ \beta_1 + R_3 3\ \beta_3 + R_4 3\ \beta_4 + D(3)^- - D(3)^+ = G(3)$$

For simplicity, these discrepancies were placed in only one priority level (F_1) and were given equal weightings ($P_i = 1.0$).

The measure of system response to user requests was taken to be response time. Thus, each $R_j i$ is in time units and G_i likewise. A specific goal of 3000 milliseconds (3 seconds) was chosen as the initial values for each G_i. The β bounds on the range of β_j's were arbitrarily set at .2 and 5.0 (L_i and U_i, respectively). The resulting MGP formulation is:

$$\text{.E.}\qquad \text{Minimize} = F_1\left(\sum_{i=1}^{3}(D_i^- + D_i^+)\right) + F_2\left(\sum_{i=1}^{4}(e_i^+ + e_i^- + \ell_i^+ + u_i^-)\right)$$

$$\text{s.t.}\qquad R \cdot \beta + D^- - D^+ = G$$

$$\beta + E^- - E^+ = \bar{1}$$

$$\beta + L^- - L^+ = L$$

$$\beta + U^- - U^+ = U$$

$$\text{where } R = \begin{bmatrix} R_1 1 & R_2 1 & R_3 1 & R_4 1 \\ R_1 2 & R_2 2 & 0 & 0 \\ R_1 3 & 0 & R_3 3 & R_4 3 \end{bmatrix} \begin{bmatrix} \beta = \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix} G = \begin{bmatrix} 3000 \\ 3000 \\ 3000 \end{bmatrix} L = \begin{bmatrix} .2 \\ .2 \\ .2 \\ .2 \end{bmatrix} U = \begin{bmatrix} 5 \\ 5 \\ 5 \\ 5 \end{bmatrix}$$

VI. Interpretation and Use of Methodology

The Goal Evaluation Stage procedures produce three sets of measures to aid the analyst in the Design Evaluation Stage. They are current design evaluation, design alternative identification and design alternative evaluation. Each set of measures individually provide insight into the performance of the system, but when viewed collectively, they provide a systematic approach to information system design.

VI.A. Current Design Evaluation

The first set of measures produced by this methodology evaluates the current information system design with respect to the individual user goals and the entire goal structure. To produce such an evaluation, the ancillary constraint of $\beta + E^- - E^+ = \bar{I}$ is assigned top priority in the goal hierarchy. The effect is to allow the formulation to yield $\beta_j = 1$ for all activities. This essentially sets the formulation to the current configuration (since $\beta_j = 1$ implies that the required level of performance equals the current level) and results in the determination of $D^+(i)$ and $D^-(i)$ for all goals in the current design.

This evaluation of individual goals is then used to evaluate the global objective function. This objective function can be a combination of goals at different priority levels. Thus, the results are a vector of collective goals' measures, with an entry for each priority level, $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ for k priority levels. (Note: For the remainder of this paper, all references to the objective function

will be denoted λ, with the understanding that the context can apply to each individual priority level, λ_k).

VI.B. Design Alternative Identification

The second set of measures is the solution vector of β_j's. They are called decision or control variables to emphasize the fact that they represent the aspects of the information system design under control of the designer/analyst. Implicit in the assumption of linear performance in this paper, is the further assumption that an activity is an indivisible unit. For example, one does not use half a compiler or one and a half job schedulers. The analyst has under his control, however, the operating characteristics of the activity, e.g. its queue discipline, number of servers, type of function, etc. Therefore, in order to achieve the indicated $(\beta_j)*100\%$ performance level for an activity, the analyst can modify the operating characteristics of that activity.

Thus β_j is interpretted as the level of performance of activity j required to satisfy the objective function, relative to the current level of performance. In other words, $\beta_j = .75$ implies that 75% of the current level must be achieved. If time is the measure, this translates into making design modifications to the activity such that the resulting performance level is reduced by one fourth. Similarly, for a β_j greater than one, the resultant change to the operating characteristics of activity j should increase the performance level.

The MGP formulation permits all β_j's to be changed simultaneously. Thus, although an activity may have been employed by many different

application services, the value of β_j produced by the MGP solution is with respect to all these associated services. The effect of changing the current level of performance for an activity on the objective function, i.e., its effect on different priority levels and penalties, has been automatically accounted for by MGP procedures.

Therefore, the output of the MGP problem formulated in .E. is a vector of β values, interpretted as indicators of the levels of performance required to satisfy all user goals. The Design Evaluation Stage uses these values to determine which activities to change ($\beta_j \neq 1.0$) and which not to change ($\beta_j = 1.0$), and to determine the direction of that change (the β_j value itself). The determination of what specific change to make is make in conjunction with outputs from the System Evaluation Stage and analyst experience.

VI.C. Design Alternative Evaluation

The third set of measures evaluate design alternatives indicated by the β_j values of the second step. The result of this evaluation is a list of the activities ordered by their ability to improve performance. This ordering is based on the marginal contribution of each activity j to the minimization of overall system performance λ denoted as $\delta\lambda_j$, given that the activity was modified according to its β_j value.

The calculation of $\delta\lambda_j$ is made for each activity defined for the current system. Based on the value of $\delta\lambda_j$ an ordered list can be assembled, ranging from the most negative value to the most positive value. The most negative value indicates that modifying the activity associated with β_j offers the possibility for the greatest reduction in discrepant system performance. The activity at the other end of the

list offers the possibility for the least reduction in λ, possibly increasing it if $\delta\lambda_j > 0$. If there are no $\delta\lambda_j$ values less than 0, then there may be no activities that can be changed to reduce λ.

This is very useful to the designer, because between iterations of a given design every desired design change cannot be made. In addition to insufficient time to implement all changes, the cost will probably be prohibitive and the ability to isolate the effect of an individual change will become extremely difficult. Thus, the designer would greatly benefit from having a facility to rank design alternatives in the order of their performance improvement capabilities.

VII. Analysis of Example System

This section serves two purposes. First, it describes the application of this methodology to the examle information system defined in Section V. Second, it provides an opportunity to discuss the advantages of this methodology over the other approaches described in Sections II and III with respect to multiple criterion information system design problems.

For the System Evaluation Stage, an IPSS[2] simulation model of this system was built. Executions of this model produced not only normal system statistics (resource queueing and utilization) but also the R matrix. This matrix was then input to the Goal Evaluation Stage, which employs the MGP based design analysis routines. Depending on the results of this analysis, the original IPSS model was modified to reflect the indicated change, and then re-input for a system evaluation, completing the iteration.

Although this methodology is not limited to using simulation for the System Evaluation Stage, it demonstrates the class of systems that can be analyzed by this methodology. A given system can be as broad in scope as to range from external user arrivals to low level access of data elements. Such a system would normally exceed the tractable bounds of the 'system oriented' approaches in Section II. Furthermore, the sequence of interrelated activities, i.e., a sequence of queueing subsystems, in this model make the dynamic interactions between activities

---

[2]The Information Processing System Simulator (IPSS) is a special purpose, discrete event simulator the development of which was conducted with the support of the National Science Foundation, initially under Grant No. GN-36622 and was continued under SIS75-21643. (See DEL77a, DEL77b).

quite significant. Both the system and user oriented approaches black box such interactions, and, thus, for the class of models used in this example, could ignore crucial performance relationships.

The results of the simulation execution are input to the MGP procedures to evaluate the example model formulated in section V. This formulation, in itself, represents an advantage of the MGP based approach over other multiple criterion methods. The weighting approach can not accomodate a goal hierarchy of more than one level. But, even in this simple model, as in all models generated by this methodology, at least two priortity levels are specified (at least one for user goals and one for β feasibility constraints). Thus, the weighting approach could not formulate this approach directly. The spatial proximity approach depends greatly on visual interpretation and since this model has five activities (variables), analysis of a fifth dimensional graph has little chance for success.

Given formulation .E. of the example model, Table 2 presents the results of the first two evaluation iterations. As described in the previous section, the application of the Goal Evaluation Stage produces three sets of measures. The evaluation of the current design for iteration 1 yields a measure of total discrepant performance, λ, (2406 time units in this case). The individual discrepancies show that for user types 1 and 3, the current system design took a longer than desired time to respond to the request, whereas the system responded to user type 2, faster than desired. In a strict linear programming approach, one constraint in three not satisfied would result in an infeasible solution. The current approach, however, allows, in fact is designed to handle, the non-optimal situation.

	Iteration 1	Iteration 2

INPUT DATA:*

$$R = \begin{bmatrix} 2280 & 40 & 44 & 1695 \\ 2770 & 70 & 0 & 0 \\ 2403 & 0 & 37 & 1747 \end{bmatrix} \qquad R = \begin{bmatrix} 2218 & 36 & 41 & 404 \\ 2352 & 64 & 0 & 0 \\ 2298 & 0 & 34 & 424 \end{bmatrix}$$

A. CURRENT DESIGN EVALUATION

λ=2406		λ=1128	
$D^+(1) = 1059$	$D^-(1) = 0$	$D^+(1) = 0$	$D^-(1) = 301$
$D^+(2) = 0$	$D^-(2) = 160$	$D^+(2) = 0$	$D^-(2) = 583$
$D^+(3) = 1187$	$D^-(3) = 0$	$D^+(3) = 0$	$D^-(3) = 244$

B. DESIGN ALTERNATIVE IDENTIFICATION

$\beta_1 = 1.005$	$\beta_1 = 1.20$
$\beta_2 = 3.31$	$\beta_2 = 2.74$
$\beta_3 = 1.00$	$\beta_3 = 1.00$
$\beta_4 = 0.33$	$\beta_4 = 0.49$

C. DESIGN ALTERNATIVE EVALUATION

$\delta\lambda_1 = -10$	$\delta\lambda_1 = -657$
$\delta\lambda_2 = -66$	$\delta\lambda_2 = -174$
$\delta\lambda_3 = 0$	$\delta\lambda_3 = 0$
$\delta\lambda_4 = -2153$	$\delta\lambda_4 = 422$

*For each and all iterations, several, not one, simulation runs were made in order to give more consistent, reliable R_{ij} values. Thus, each R_{ij} value is a mean value across simulation executions, not a point-estimate of only one particular execution.

Table 2: Raw and Calculated Data for Example System

Examination of the β_j's, however, demonstrates the insight available through this methodology. Total approaches could easily evaluate the current design, i.e., determine the discrepancies and λ. But the ability to relate this aggregate performance to possible design modifications is not an integral part of the total approach. As will be seen below, the β_j values resulting from the solution of the associated MGP formulation can provide such insight.

For the example system in iteration 1, the values of $\beta_3 = 1.0$ and $\beta_1 = 1.005$ imply that for the current system, activities A_1 and A_3 are performing at or very close to satisfactory levels. Activities A_2 and A_4 on the other hand, are performing at very unsatisfactory levels. Since the individual β_j value indicated in which direction a change should be made, A_2 should be lengthened (i.e., it currently contains slack) and A_4 needs to be reduced. But the relative distance a particular β_j is from the normal value of 1.0 does not alone provide enough information to determine which activity to modify first.

The values of $\delta\lambda_j$, however, provide such information. A_3, because it is performing at a satisfactory level, need not be changed, and there would not be any marginal contribution to reducing λ (hence, $\delta\lambda_3 = 0.0$). Similarily, β_1 for A_1 implies only a small change and, thus, $\delta\lambda_1$ is also small. The main decision then for the designer in iteration is to decide which of the remaining activities to modify, A_2 or A_4. Comparing $\delta\lambda_2$ and $\delta\lambda_4$ provides definite guidance. A_4 should be modified, and since $\beta_4 < 1$, it should reduce its current performance level.

This guidance was followed in the model in iteration 2. Activity A_4 was modified to reduce its performance level (processing time)

significantly. The effect can initially be seen by comparing the R matrices for iterations 1 and 2. The column of $R_4 i$ is greatly reduced in iteration 2 from that in iteration 1. λ has also been greatly reduced which implies that performance was indeed improved by following the β_j and $\delta\lambda_j$ interpretations. The individual discrepancies show a shift to a design which satisfices all responses too fast (on the average). Analysis of the β_j and $\delta\lambda_j$ measures now reveal that the burden of performance improvement has been shifted to activity A_1.

This process of basing design decisions on $\delta\lambda_j$ and β_j was continued for several more iterations. The design modifications indicated in iteration 2 (make A_1 longer) was followed and overall λ was reduced to 500 time units from 1128. Figure 7 shows how over the entire set of iterations for this model, use of $\delta\lambda_j$ and λ_j consistently reduced λ. Two other information systems were simulated and analyzed using this methodology. Applying the same $\delta\lambda_j/\beta_j$ based decision rules, they achieved markedly similar effects on λ over a sequence of design decisions. The results of these analyses are presented in full in CHN77.

The freedom of direction in design represented by the $\delta\lambda_j/\beta_j$ analysis is an advantage over the global approaches. The measure of performance (in this case, time) does not have to be transformed to some meaningless aggregate measure for the sake of tractability. Other global approaches require an a priori set of options from which design alternatives are selected. This methodology, however, allows the entire set of design alternatives to be available throughout the design process. Through appropriate analysis of the β_j's and λ, on a given iteration, this set is reduced to the most beneficial subset for consideration by

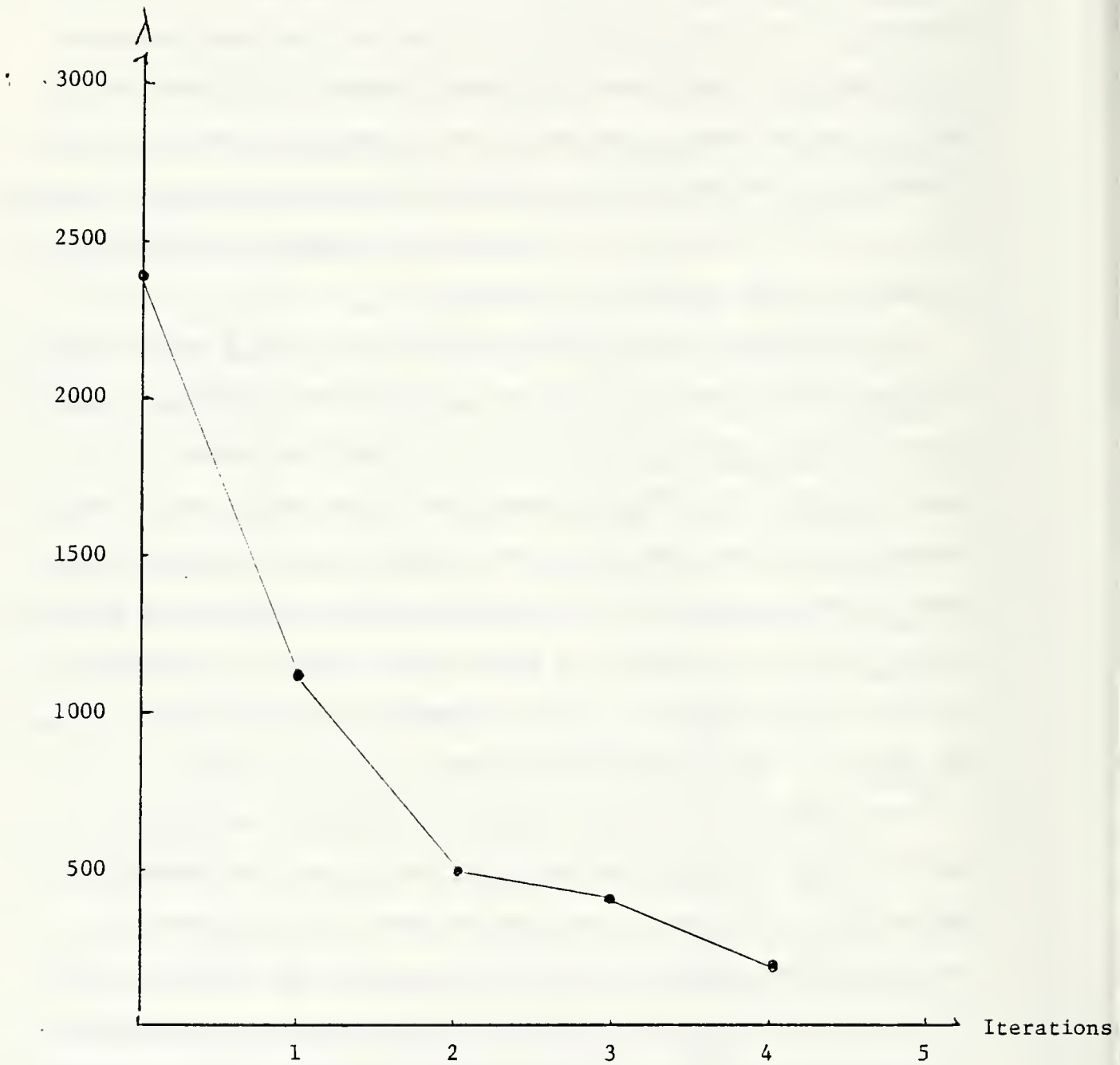

Figure 7 Results of Design Decision Sequence

the designer. On subsequent iterations, however, the complete set of alternatives is available again. Such flexibility provides for continued creativity for the designer and lessens the chance of locking the designer into an unfruitful path because of errors in design early in the sequence of design iteration.

VIII. Summary and Extensions

The goal of the research reported in this paper is to aid the analyst in the design and analysis of computer-based information systems. To that end, this research accomplished the following:

1. the development of a general, multi-stage, iterative methodology for the evaluation of systems with respect to multiple criteria,
2. the establishment of a formal liaison between the attainment of user goals and the utilization of resources,
3. the development of an evaluation method for identifying information system performance levels, and
4. the development of predictive measures of future system performance for evaluating design alternatives.

To demonstrate the practicality of this research, the methodology was implemented as an adjunct to IPSS. This realization was verified and validated in a two phase process. The goal of Phase 1 was to demonstrate both the feasibility and validity of the Stage 2 formulations. This was accomplished through a series of experiments where design alternative selections using Stage 2 analysis were tested. The result was that design selections guided by β_j and $\delta\lambda_j$ values had consistently better performance than selections made without the use of β_j and $\delta\lambda_j$. In phase 2, the realization in IPSS of the statistical liaison between the System Evaluation Stage and the Goal Evaluation Stage was verified. When the new automatic IPSS statistics were compared with those collected in Phase 1 they were found to be identical.

The overall significance of these verification and validation exercises is to demonstrate the feasibility and validity of the methodology developed in this research. The procedures developed for the Goal Evaluation Stage were the main focus of this research and were shown, by the experiments, to provide consistent guidelines for making design selections to improve performance. The realization in IPSS of the extension was shown to be not only feasible but also beneficial to the modeling process. Thus, the multi-stage, multi-criterion methodology developed in the research is a useful evaluative tool for the designer.

One important aspect of this research is its extensibility. There are many areas of potential research in both information system's and general system's design and evaluation. The methodology and measures developed in this research provide a framework for modeling and analyzing such systems. The current realization of this methodology can be extended to increased analysis. Additional experimentation in information systems with more complex goal structures or more complex system activity or both should be examined. An existing man-machine information system is currently being modeled to analyze these areas.

Mean response time was used in the current research as the measure of activity performance and goal attainment. As has been noted however, the methodology was designed to be applicable to any measurable performance criterion. Thus, an area of future research is to use this methodology in the investigation of information system performance measures. An appropriate first choice is to study the measure of cost. Other possibilities include throughput, resource utilization,

reliability and user satisfaction. The sensitivity of the model parameters and measures to perturbations in performance can also be investigated. The evaluative environment provided by this methodology, in particular the model modification capabilities of IPSS, allows one to easily vary the parameters of a model and collect comparative statistics.

The modeling view and interpretation this methodology takes of an information system is not restricted to information systems alone. The concept of a system as an interrelated collection of services and activities is easily applied to any system, particularly ones that take the form of general job-shop. Thus, a similar MGP formulation can be constructed to analyze the performance of any such system and the analysis and measures developed in this research can be applied to the resultant MGP solution. In this manner, performance sensitive activities within a general system can be identified and modifications to their performance evaluated.

Current research is being conducted on applying this methodology to the design and evaluation of internal accounting control features. In this case user performance goals are not only time and cost, but also hazard exposure and system reliability. The resultant methodology will be used to evaluate the cost/effectiveness of features, to aid in determining feature placement for minimum cost and to aid in determining audit scope for external auditors.

IX. References

ABA68 Abate, J., Dubner, H., and Weinberg, S.B., "Queueing Analysis of IBM 2314 Disk Storage Facility," Journal of Association for Computing Machinery (JACM), Vol. 15, No. 4, (1968), pp. 577-589.

ADI72 Adiri, Igal, "Queueing Models for Multiprogrammed Computers," IBM Research Report - RC3802 (March 30, 1972).

BAR71 Bard, Y., "Performance Criteria and Measurement for a Time-Sharing System," IBM System Journal, No. 3, (1971), pp. 193-216.

BAS75 Baskett, F., Chandy, K., Muntz, R., and Palacaios, F., "Open Closed and Mixed Networks of Queues with Different Classes of Customers," JACM, Vol. 22, No. 2, (1972), pp. 248-260.

BET71 Bettman, J., "A Graph Theory Approach to Comparing Consumer Information Processing Models," Management Science, Vol. 18, (1971), pp. 114-129.

BOE75 Boehm, B., and Bell, T., "Issues in Computer Performance Evaluation — Some Consensus, Some Divergence," Performance Evaluation Review (PER), Vol. 4, No. 3, (1975), pp. 4-40.

BOU72 Boutross, King and Rutledge, "Discrete Simulation Model of "ADSD," IBM Research Report - RC3698, (January 20, 1972).

BUZ73 Buzen, J.P., "Computational Algorithms for Closed Queueing Networds with Exponential Servers," Communications of the ACM (CACM), Vol. 16, No. 9, (1973), pp. 527-531.

BUZ76 Buzen, J. P., "Fundamental Laws of Computer System Performance," Conference on Performance, Modeling and Measurement Evaluation (CPMME), (1976), pp. 200-210.

CHN77 Chandler, John S., "A Methodology for Identifying Information System Design Requirements Based on the Assessment of Multiple User Performance Criteria," unpublished Ph.D. Dissertation, The Ohio State University, 1977.

CHA61 Charnes, A., and Cooper, W.W., Management Models and Industrial Applications of Linear Programming, Wiley & Sons, New York, (1961), pp. 215-223.

CHA68 Charnes, A., and Cooper W.W., "A Goal Programming Model for Media Planning," Management Science, Vol. 14, No. 8, (1968), pp. 423-430.

CHA70 Charnes, A., Cooper, W.W., Neihaus, R.J., and Sholtz, D., "A Model for Civilian Manpower Management in the U.s. Navy," Models of Manpower Systems, A.R. Smith (Ed.), English University Press (1976).

CHA76 Charnes, A., Cooper, W.W., Lewis, K.A., and Niehaus, R.J., "A Multi-objective Model for Planning Equal Employment Opportunities," Multiple Criteria Decision Making (MCDM), KYOTO 1975, Springer-Verlag, Berlin, (1976), pp. 111-134.

COL72 Cole, Gerald D., "Performance Measurements on the ARPA Computer Network," IEEE Transactions in Communication, (1972), pp. 20-23.

COO72 Cooperman, J.A., Lynch, H.W., and Tatzlaff, W.M., "SPG: An Effective Use of Performance and Usage Data," COMPUTER (1972), pp. 20-23.

DEL77a DeLutis, T.G., "The Information Processing System Simulator (IPSS): Language Syntax and Semantics," unpublished research report (Grant No. GN36622), March, 1977.

DEL77b DeLutis, T.G., "A Methodology for the Performance Evaluation of Information Processing Systems," Final Report to National Science Foundation, CISRC Technical Report OSU-CISRC-TR-77-2.

GAV76 Gaver, D.P., Jr., and Humfeld, George, "Multitype Multiprogramming: Probability Models and Numerical Procedures," CPMME, (1976), pp. 38-43.

GOT73 Gotlieb, C.C., and MacEwen, G.H., "Performance of Movable-Head Disk Storage Devices," JACM, Vol. 20, No. 4, (1973), pp. 604-623.

GRC72 Grochow, Jerald, "Utility Functions for Time-Sharing System Performance Evaluation," COMPUTER (September/October 1972), pp. 16-19.

HAL74 Hall, W.A., "A Simulation Model to Aid in the Design and Tuning of Hierarchical Databases," WSC, (1974), pp. 277-284.

HEL70 Hellerman, H., and Smith, H.J., Jr., "Throughput Analysis of Some Idealized Input, Output and Computer Overlap Configurations," Computing Surveys, Vol. 2, No. 2 (1970), pp. 111-118.

IGN76 Ignizio, J.P., Goal Programming and Extension, D.C. Heath, Lexington, Massachusetts, (1976).

IJI65 Ijiri, Y., Management Goals and Accounting for Control, Rand McNally, Chicago, (1965).

JAA69 Jaaskelainen, V., "A Goal Programming Model of Aggregate Production Planning," Swedish Journal of Economics, Vol. 71, #2 (1969), pp. 14-29.

JOS68 Joslin, E.V., Computer Selection, Addison Wesley Publishing Company, Inc., Reading Massachusetts, (1968).

KAC74 Kacchal, S., "The Configurational Optimization of Computer Systems," Ph.D. Dissertation, University of Minnesota, 1974.

KIL73 Killough, Larry N., and Sanders, Thomas L., "A Goal Programming Model for Public Accounting Firms," Accounting Review, XLVIII, #2, (April, 1973), pp. 268-279.

KLE76 Kleinrock, Leonard, Queueing Systems - Volume I: Theory, Volume II: Computer Applications, John Wiley & Sons, New York, (1976).

KOL73 Kolence, K., "The Software Empiricist," PER, Vol. 2, No. 2, (1973), pp. 31-36.

LEE71 Lee, S., and Jaaskelainen, V., "Goal Programming: Management's Math Model," Industrial Engineering, February, 1971, pp. 30-35.

LEE72 Lee, S., Goal Programming for Decision Analysis, Auerbach, Philadelphia, (1972).

LIN76 Lindsay, David S., "A Hardward Monitor Study of a CDC KRONOS System," CPMME, (1976), pp. 136-144.

MAC73 MacCrimmon, R.K., "An Overview of Multiple Objective Decision Making," MCDM, (1973), pp. 18-44.

MER74 Merrill, H.W.B., "In Using Statistical Analysis of SMF Data to Build Usable Simulation Models to Large-Scale Computer Systems," Symposium on the Simulation of Computer Systems - II (SSCSII), (1974), pp. 181-187.

MOR74 Morris, Michael F., "Kiviat Graphs -- Conventions and Figures of Merit," PER, Vol. 3, No. 3, (1974), pp. 3-7.

NAH74 Nahouraii, E., "Direct Access Device Simulation," IBM Systems Journal, vol. 13, No. 1, (1974), pp. 19-31.

NOR71 Norland, K.E., and Bulgren, W.C., "A Simulation Model of GECOS III," ACM, (1971), pp. 596-612.

NUN71 Nunamaker, J.F., "A Methodology for the Design and Optimization of Information Processing Systems," AFIPS SJCC, Vol. 38, (1971), pp. 283-294.

NUN76 Nunamaker, J.F., Konsynski, B., Ho, T., and Singer, C., "Computer-Aided Analysis and Design of Information Systems," CACM, Vol. 19, No. 12, (1976), pp. 674-687.

REI76 Reiter, A., "Some Experiments in Directory Organization -- A Simulation Study," CPMME, (1976), pp. 1-9.

ROS74 Rose, L.L., Gotterer, G.A., Hayya, J.C., "A Simulation Model for Dynamic File Management," WSC, (1974), pp. 251-257.

SCH72 Scharf, T., "Weighted Ranking by Levels," in Computer Evaluation and Methods, P. Hildas (Ed.), Universitetsforlaget Oslo, (1972).

SHW65 Schneidewind, N.F., "Analytical Model for the Design and Selection of Electronic Digital Computing Systems," D.B.A. Dissertation, Universityof Southern California, 1965.

SEV72 Severance, Dennis G., and Merten, Alan G., "Performance Evaluation of File Organization Through Modeling," ACM, (1972), pp. 1061-1072.

SIL74 Siler, K.F., "A Model for the Evaluation of Large Scale Data Retrieval Systems," WSC, (1974), pp. 259-274.

SVO76 Svobodova, Liba, Computer Performance Measurement and Evaluation Methods: Analysis and Applications, American - Elsevier, New York, (1976).

TEI67 Teichrow, D., "ISDOS - A Research Project to Develop a Methodology for the Automatic Design and Construction of Information Procesing Systems," ISDOS Working Paper 1, No. 1, Case Institute of Technology, (August 1967).

BOUND BY
HV
3-94

UNIVERSITY OF ILLINOIS-URBANA
3 0112 039809931